[20].	 a) repeat 8 first s b) repeat that fo c) find the best of c/ul> I will start with start 	steps from par or many other one, tune, try t	rt 1 for new c regression m to explain;	odels;	ataset							
[28]:	<pre># This is secon # imports import pandas a from sklearn.mo from sklearn.pr from sklearn.me from sklearn.mo from sklearn.mo import numpy as from sklearn.li from sklearn.tr from sklearn.en from sklearn.ne from sklearn.ne from sklearn.ne from sklearn.ne</pre>	del_selecti reprocessing m import SV etrics import del_selecti del_selecti np near_model ree import D semble import iye_bayes i peline impore	on import import St con import on import import Log ecisionTre ort RandomF ort KNeigh import Gaus ort Pipelin	train_tes andardSca cation_re cross_val GridSearc isticRegr eClassifi orestClas borsClass sianNB e	et_split eler eport, cor escore chcv eession eer esifier				acy_sco	re		
[29]:		ead(10))		ss Unifor 5 5	emity of (Cell Si	ze \ 1 4					
	2 10 3 10 4 10 5 10 6 10 7 10 8 10	015425 016277 017023 017122 018099 018561 033078		3 6 4 8 1 2 2			1 8 1 10 1 1					
	9 10 Uniformity of 0 1 2 3 4	33078 E Cell Shape 1 2 1 8	e Marginal L 1 L	Adhesion Adhesion 1 5 1	n Single	Epithe	2 lial Cel	ll Size 2 7 2 3 2				
	5 6 7 8 9 Bare Nuclei	1 1	L 2 L	8 1 1 1 1 nal Nucleo		ses Cl		7 2 2 2 2				
	0 1 1 10 2 2 3 4 4 1 5 10 6 10 7 1		3 3 3 3 9 3		1 2 1 7 1 7 1	1 1 1 1 1 1	2 2 2 2 2 4 2					
[30]:	<pre>8</pre>	taset.drop('Class', a		1	5 1	2 2	datafra.	mes.			
[31]: [32]:	<pre># another advic standard_scaler scaled_matrix = # split into tr</pre>	= Standard standard_s	Scaler()			pefore	splittir	ng it i	nto tra	in and t	est varia	bles
[33]:	<pre># create SVC, m svc_classifier svc_classifier. svc prediction</pre>	nake a predi SVC (kerne fit (X_train	y_test = t .ction el='linear' n, y_train)	, random_	state = (atrix, y	, test	_size=0	.2)		
t[33]:	svc_prediction # well, not exa array([2, 4, 2, 2, 2, 4, 2, 2, 4,	ectly the wa	2, 2, 2, 4, 2, 4, 2, 4, 4,	2, 2, 2, 2, 2, 2, 2, 4, 2,	2, 2, 4, 2, 4, 4, 2, 2, 4,	4, 2, 2, 4,	2, 4, 2, 2, 2, 4,	, 2, , 4,				
[34]:	2, 2, 4, 4, 4, 2, 2, 2, 2, # compare predi print(confusion print(classific	4, 4, 2, 4, 2, 2, 2, 2, 2, 2], dtyp 	2, 4, 4, 2, 2, 4, pe=int64) actual restest, svc_pret(y_test, s	2, 4, 2, 2, 2, 2, sults ediction) vc_predic	2, 2, 4, 2, 2, 2,)	2, 2, 4, 2,	2, 4, 2, 4, 2, 2,	, 2,				
	<pre># results diffe # and in genera # # Also this is # train and tes print(accuracy_ [[92 0] [0 45]]</pre>	worth menti t I was get score(y_tes	od enough Coning that Cting worse	without results diction))	'fit and than here	transf						
	accuracy macro avg weighted avg	1.00	1.00	1.00 1.00 1.00 1.00	45 137 137 137							
[35]:	<pre># now we will c scores = cross_ print(scores) print(scores.me # Ugh, it seems</pre>	val_score(s an()) we were ju	vc_classif ust very lu	ier, scal	ed_matri>	к, у, с	v=10)	accura	cy			
[36]:	[0.92753623 0.97 0.95588235 1. 0.96638959931798 It already looks that t	0.97 882 this dataset is grid search	much more f	98529412]				edictions	, but let's	try other	models too	
	<pre># set C paramet param_grid = [{'C': [0.25,] grid_search_res grid_search_res</pre>	0.5, 0.75, ult = Grids	1], 'kerne	c_classif			, scorir	ng="acc	uracy",	n_jobs=	-2)	
	<pre>print("Best par print() print("Grid sco print() means = grid_se stds = grid_sea</pre>	eres on deve	elopment se	t:") s_["mean_	_test_scor	ce"]	rid_sear	cch_res	ult.bes	t_params	_))	
	best_re	through seven params in z to < mean): tro = mean sult['mean'	<pre>cip(means,] = mean</pre>		d_search_	_result	.cv_resu	ılts_["	params"]):		
	print("%0.3 print() print('Best res # to be fair, a print("classifi print() y true, y pred	ults found 11 3 rbf ke	with %0.3f ernels show ort:")	% (mean, mean and red the sa	l paramete nme result	ers %s'		c_resul	t['mean	'], best	_result['	params'])
	print (classific Best parameters Grid scores on d 0.956 (+/-0.056)	set found of	t(y_true, on developm	y_pred))	{'C': 0.	-	rnel':	'rbf'}				
	0.956 (+/-0.035) 0.956 (+/-0.052) 0.956 (+/-0.042) 0.956 (+/-0.052) 0.960 (+/-0.041) 0.956 (+/-0.052) 0.958 (+/-0.042)	<pre>for { 'C': for { 'C': for { 'C': for { 'C': for { 'C': for { 'C': } }</pre>	0.25, 'kern 0.5, 'kern 0.5, 'kern 0.75, 'kern 0.75, 'kern 1, 'kernel	nel': 'rb nel': 'lin nel': 'rb nel': 'rb nel': 'li nel': 'rb	of'} near'} inear'} of'} ar'}							
	Best results fou classification r	report:	960 mean an recall f1	_	ters {'C' support 92 45	: 0.75,	'kernel	l': 'rb	f'}			
	macro avg weighted avg This accuracy looks a outstanding, no matt Now I need to imple 1. Logistic Regressi 2. Decision Tree Cla 3. Random Forest Cla 4. K- Nearest Neight 5. Naïve Bayes	ter what paran ment a lot of o ion assifier Classifier (with	neters we use	dels	137 137 I before, so	there is	nothing t	o be sur	prised ab	out, I gue:	ss. It is in ge	eneral
[37]:	# 1. Logistic r logistic_regres logistic_regres	sion = Logi sion.fit(X_	sticRegres train, y_t	rain)								
	# 1. Logistic r logistic_regres	regression sion = Logi sion.fit(X_ gistic_regr val_score(l curacy with	sticRegres train, y_t ression.pre ogistic_re a std of %0	rain) dict(X_te gression, .2f" % (s	scaled_n scores.mea	an(), s	cores.st	zd()))	ode optir	nized, so ł	nere is my g	jeneric
	# 1. Logistic r logistic_regres logistic_regres prediction = lo scores = cross_ print("%0.2f ac 0.97 accuracy wi Making regression m approach: best_result = { mean_zero = 0 def generic_reg global mean global best regression_ prediction print(regre print(confu print(regre print(class scores = cr mean = score if(mean_zer mean_ze best_re best_re	regression sion = Logi sion.fit(X_ gistic_regreval_score() curacy with the std of () nodels one by ression_model zero result model.fit(X_ = regression_model sion_matrix ssion_model ification_ression_model ification_ression_std() co < mean): ro = mean sult['regressult['accuracy testion_model sistion_model sult['regressult['score sult['score	sticRegres train, y_t ession.pre ogistic_re std of %0 0.02 one like wha del_executi (_train, y_ on_model.pr _name,": (y_test, p _name,": report(y_te ore(regress ession_mode eacy'] = me = std _name,":	on (regres train) edict(X_te gression, .2f" % (s tyou see ab on (regres train) edict(X_t confusion rediction classific st, predi ion_model el_name'] an %0.3f acc	scaled_n cores.mea cove is cute ssion_mode est) matrix ' n) cation rep ction)) , scaled_ = regress curacy with	el, regulative matrix sion_modeth std	ant to mare ression_ del_name of %0.3f	_model_:	name): ean, st	d))	nere is my g	generic
	# 1. Logistic r logistic_regres logistic_regres prediction = lo scores = cross_ print("%0.2f ac 0.97 accuracy wi Making regression m approach: best_result = { mean_zero = 0 def generic_reg global mean global best regression_ prediction print(regre print(confu print(regre print(class scores = cr mean_ze best_re best_re best_re best_re print(regre print(regre print(regre print(regre print(regre print(regress) ist_of_regress list_of_regress	regression sion = Logi sion.fit(X_ gistic_regression_score(I curacy with the std of Condels one by ression_models condels one by ression_model sion_models fication_ression_model fication_ress_val_scores.mean() s.std() co < mean): result['regrees: sult['regrees: sult['std'] ssion_models fion_models fion_	sticRegres train, y_t ession.pre ogistic_re std of %0 0.02 one like wha del_executi (_train, y_ on_model.prname,": (y_test, pname,": eport(y_te ore(regress ession_mode acy'] = me = stdname,": 'LogisticRegress' 'LogisticRegress' 'SupportVegregression del_executi del is %s w	on (regression, redict (X_temperature) on (regression) edict (X_temperature) train) edict (X_temperature) confusion classific st, prediction classif	scaled_n scores.mea cove is cute ssion_mode est) matrix ' sation rep ction)) , scaled_ = regress suracy wit ===================================	el, region model std elic Regresion Random Finbors Cla	cores.st ant to ma ression_ del_name of %0.3f ession() TreeClast orestClast assifier	model_: =10) =10) ==10; ==10; ========================	ean, st -=-=- () r(n_est est_res est_res	d)) ') imators=	10) ression_muracy'],	
	# 1. Logistic r logistic_regres prediction = lo scores = cross_ print("%0.2f ac 0.97 accuracy wi Making regression m approach: best_result = { mean_zero = 0 def generic_reg global mean global best regression_ prediction print(regre print(confu print(regre print(class scores = cr mean_zer best_re best_re	regression sion = Logi sion.fit(X_ gistic_regression_score() curacy with the std of () nodels one by ression_model sion_model.fit(X_ ssion_model sion_models ification_ression_model sion_models ification_ression_scores.mean() ss.std() co < mean): ro = mean sult['regression_scores.mean() ss.std() co < mean): ro = mean sult['regression_models ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_models[ion_model	sticRegres train, y_t ession.pre ogistic_re std of %0 0.02 one like wha del_executi (_train, y_ on_model.pr _name,": (y_test, p _name,": (y_test, p _name,": (y_test) report(y_te ere(regress ession_mode acy'] = me = std _name,": "LogisticRegress 'LogisticRegress 'NaiveByte 'SupportVe regression del_executi del is %s we dision matri esification	crain) dict(X_tegression, .2f" % (s tyou see at on (regres train) edict(X_t confusion rediction rediction classific st, predi ion_model an %0.3f acc reeClassi restClass Neighbors styl=Gauss styl=Gauss ctorMachi on(value, rith accur	scaled_n scores.mea cove is cute ssion_mode est) matrix ' sation rep ction)) , scaled_ = regress suracy wit ===================================	el, region model std elic Regresion Random Finbors Cla	cores.st ant to ma ression_ del_name of %0.3f ession() TreeClast orestClast assifier	model_: =10) =10) ==10; ==10; ========================	ean, st -=-=- () r(n_est est_res est_res	d)) ') imators= ult['regult['acc	10) ression_muracy'],	
	# 1. Logistic r logistic_regres prediction = lo scores = cross_ print("%0.2f ac 0.97 accuracy wi Making regression m approach: best_result = { mean_zero = 0 def generic_reg global mean global best regression_ prediction print(regre print(confu print(class scores = cr mean_ze best_re print(regre print(regre print(regre print(regre print(regre print(regre print(regre print(regre print() print('= print() list_of_regress list_of_reg	regression sion = Logi sion.fit (X_ gistic_regreval_score(I curacy with the std of Condels one by ression_model zero result model.fit (X_ regression_model sion_models ification_ression_model sion_models result['regrees.mean() s.std() rescon_models sion_models ion_models ion	sticRegres train, y_t ession.pre ogistic_re std of %0 0.02 one like wha lel_executi (_train, y_ on_model.pr _name,": (y_test, p _name,": eport(y_test) report(y_test) rep	crain) dict(X_te gression, .2f" % (s tyou see at on (regres train) edict(X_t confusion rediction classific st, predi ion_model an %0.3f acc "reeClassi restClass Neighbors storMachi con(value, ith accur accur .x reportscore 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.	scaled_n scores.mea cove is cute ssion_mode est) matrix ' stion rep ction) scaled_ = regress curacy wit ===================================	an(), so h, but wo el, reg ') matrix sion_mod th std cicRegre ecision chorscla	ant to ma ression_ del_name of %0.3f ession() TreeClase assifier td %0.3f	model_: ssifier sc() "% (b) b	ean, st -=-=- () r(n_est est_res est_res	d)) ') imators= ult['regult['acc	10) ression_muracy'],	
	# 1. Logistic regress logistic_regress prediction = loscores = cross_print("%0.2f ac 0.97 accuracy wi Making regression mapproach: best_result = { mean_zero = 0 def generic_reg	regression sion = Logi sion.fit (X_ gistic_regression_score (I curacy with the std of Condels one by ression_model sion_model.fit (X_ regression_model sion_model.fication_ression_model sion_models sion_models sion_models ion_models ion_model	sticRegres train, y_t ession.pre ogistic_re std of %0 0.02 one like wha lel_executi (_train, y_ on_model.prname,": (y_test, pname,": (y_test, pname,": eport(y_te re(regress 'LogisticR 'DecisionT 'RandomFor 'K-Nearest 'NaiveByte 'SupportVe regression lel_executi lel is %s w sification recall fl 1.00 0.98 0.99 0.99 6.accuracy enfusion m classificat confusion m classificat	crain) dict(X_te gression, .2f" % (s tyou see ab on (regres train) edict(X_t confusion rediction classific st, predi ion_model an %0.3f acc reeClassi restClass Neighbors styl=Gauss classican classific restClass Neighbors styl=Gauss class non (value, styl=Causs class non (value, styl=Causs class class non (value, styl=Causs class cla	scaled_n scores.mea pove is cute sion_mode est) matrix ' sation rep ction)) scaled_ = regress curacy wit ===================================	an(), so h, but wo el, reg ') matrix sion_mod th std cicRegre ecision chorscla	ant to ma ression_ del_name of %0.3f ession() TreeClase assifier td %0.3f	model_: ssifier sc() "% (b) b	ean, st -=-=- () r(n_est est_res est_res	d)) ') imators= ult['regult['acc	10) ression_muracy'],	
	# 1. Logistic regres logistic_regres prediction = lo scores = cross_print("%0.2f ac 0.97 accuracy wi Making regression mapproach: best_result = { mean_zero = 0 def generic_reg global mean global best regression_prediction print(regre print(confu print (regre print (class scores = cr mean = score if (mean_zer best_re best_re best_re best_re print() print('=-=- print()) print('=-=- print()) list_of_regress list	regression sion = Logi sion.fit (X_ gistic_regression_val_score() curacy with the std of (Condels one by ression_model sion_model.fit (X_ regression_model sion_models sion_m	sticRegres train, y_t ession.pre ogistic_re std of %0 0.02 one like wha lel_executi (_train, y_ on_model.pr _name,": (y_test, p _name,": (y_test, p _name,": eport(y_te ere(regress 'LogisticR 'DecisionT 'RandomFor 'K-Nearest 'NaiveByte 'SupportVe regression lel_executi lel is %s w dision matri sification recall fl 1.00 0.98 0.99 0.99 6.accuracy energial fl 1.00 0.98	crain) dict(X_te gression, .2f" % (s tyou see at on (regres train) edict(X_t confusion rediction classific st, predi ion_model an %0.3f acc egression reeClassi restClass Neighbors storMachi on(value, ith accur accur .x reportscore 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.	scaled_n scores.mea cove is cute sion_mode sest) matrix ' sation rep ction)) scaled_ = regress suracy wit = stij=Logist fier'j=F stij=KNeigh sianNB() ne']=SVC tems(): key) cacy %0.3f support 92 45 137 137 137 137 137 137 137 137	el, reg	ant to ma ression_ del_name of %0.3f ession() TreeClase assifier td %0.3f	model_: ssifier sc() "% (b) b	ean, st -=-=- () r(n_est est_res est_res	d)) ') imators= ult['regult['acc	10) ression_muracy'],	
	# 1. Logistic regres logistic_regres prediction = lo scores = cross_print("%0.2f ac 0.97 accuracy wi Making regression mapproach: best_result = { mean_zero = 0 def generic_reg	regression sion = Logi sion.fit (X_ gistic_regreval_score() curacy with the std of (Condels one by ression_model sion_model.fit (X_ regression_model sion_model.fit (X_ ssion_model.fit (sticRegres train, y_t ession.pre ogistic_re std of %0 0.02 one like wha lel_executi (_train, y_ on_model.pr _name,": (y_test, p _name,": (y_test, p _name,": eport(y_te ere(regress leacy'] = me = std .name,": "K-Nearest 'NaiveByte 'SupportVe regression lel_executi lel is %s w lision matri sification recall fl 1.00 0.98 0.99 0.99 6.accuracy energial fl 0.99 0.99 6.accuracy energial fl 0.99 0.97 0.97 0.98	main) dict(X_te gression, .2f" % (s tyou see at on (regres train) edict(X_t confusion classific st, predi ion_model an %0.3f acc egression reeClassi restClass Neighbors storMachi models.i on(value, ith accur accur .x reportscore 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.	scaled_nescores.means poove is cute sion_mode est) matrix ' stion rep ction)) scaled_ = regress curacy wite sij=Logist fier']=F sij=KNeigh sianNB() ne']=SVC tems(): key) sacy %0.3f support 92 45 137 137 137 137 std of 0	el, reg	ant to ma ression_ del_name of %0.3f ession() TreeClas orestCla assifier td %0.3f	model_: *** * (m** **ssifier **ssifie *** * (b b b	ean, st -=-=- () r(n_est est_res est_res	d)) ') imators= ult['regult['acc	10) ression_muracy'],	
	# 1. Logistic regres logistic_regres prediction = lo scores = cross_print("%0.2f ac 0.97 accuracy wi Making regression mapproach: best_result = { mean_zero = 0 def generic_reg global mean global best regression_prediction print(regre print(confu print(regre print(class scores = cr mean = score std = score if (mean_zer	regression resion = Logi resion.fit (X_ regression.fit (X_ regression.fit (X_ regression.wal_score (I recuracy with th std of (I ression.models (I) ression.model.fit (X_ ressi	sticRegres train, y_t ession.pre ogistic_re std of %0 0.02 one like wha lel_executi (_train, y_ on_model.pr _name,": (y_test, p _name,": (y_test, p _name,": (y_test) = std name,": (y_test) - core (regress lel_executi 'K-Nearest 'NaiveByte 'SupportVe regression lel_executi lel is %s w lision matri sification recall fl 1.00 0.98 0.99 0.99 6. accuracy confusion m classificat recall fl 0.99 0.87 0.93 0.95 0.947 accur confusion classificat recall fl 1.00 0.93 0.97	con (regression, 2f" % (set you see at a confusion rediction rediction rediction rediction rediction rediction rediction rediction recursion report recursion rec	scaled_nericores.mean scores.mean scores.m	el, reg	ant to ma ression_ del_name of %0.3f ession() TreeClas orestCla assifier td %0.3f	model_: *** * (m** **ssifier **ssifie *** * (b b b	ean, st () r(n_est est_res est_res	d)) ') imators= ult['regult['acc	10) ression_muracy'],	
	# 1. Logistic r logistic_regres logistic_regres prediction = lo scores = cross_ print("%0.2f ac 0.97 accuracy wi Making regression m approach: best_result = { mean_zero = 0 def generic_reg global mean global best regression print(regre print(confu print(regre print(class scores = cr mean_ze best_re print() print('= print() print('= print() list_of_regress list_of_regres	regression sion = Logi sion.fit (X_ registic_regression_real_score (1) curacy with the std of (1) the std of (2) the std of (3) the std of (4) the std of (4) the std of (5) the std of (6) the std of (6) the std of (7) the std of (7) the std of (8) the std of (8	sticRegres train, y_t ession.pre ogistic_re std of %0 0.02 one like what classion_model.pr cname,": (y_test, p cname,": (y_test, p cname,": (y_test, p cname,": (y_test) cre(regress confusion classification classifica	rain) dict(X_te gression, .2f" % (s tyou see at on (regres train) edict(X_t confusion rediction classific st, predi ion_model l_name'] an %0.3f acc segression reeClassi restClass Neighbors storMachi con(value, ith accur .x reportscore 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.9	scaled_nericores.means poove is cute sion_mode est) matrix ' sion_mode est) matrix ' stion rep ction)) scaled_ = regress curacy wite fier']=E sifier']=F sinnB() ne']=SVC tems(): key) sacy %0.3f support 92 45 137 137 137 137 137 137 137 13	an(), so , but I was el, reg ') ort ") matrix sion_mod th std cicRegre ecision RandomFonborsCl () and s	ant to ma ression_ del_name of %0.3f ession() TreeClass crestClass assifier td %0.3f	model_: ssifier ssifie ssifier ssifie ssifier ssifie ssifie ssifie	ean, st () r(n_est est_res est_res	d)) ') imators= ult['regult['acc	10) ression_muracy'],	
	# 1. Logistic regres logistic_regres prediction = lo scores = cross_ print("%0.2f ac 0.97 accuracy wi Making regression mapproach: best_result = { mean_zero = 0 def generic_reg global mean global best regression_ prediction print(regre print(confu print(regre print(class scores = cr mean_zer best_re best_re best_re best_re pest_re print(regre print() print("= print() list_of_regress list_o	regression sion = Logi sion.fit(X gistic_regression.gion_models ression_model sion_matrix sion_model.fit(X ession_model.sion_models ross_val_sco result regression.gion_models ross_val_sco result['regrees sult['regrees result['std'] ression_models	sticRegres train, y_t ression.pre ogistic_re ogistic_re std of %0 0.02 one like what classion model.pr classion model.p	rain) dict(X_te gression, .2f" % (s tyou see ab on (regres train) edict(X_t confusion rediction rediction rediction rediction rediction rediction rediction reclassific st, predi ion_model an on(value, ith accur ac	scaled_nericores.mean cores.mean cove is cute sion_mode est) matrix ' sion_mode est) matrix ' cition) scaled_ = regress curacy with ===================================	an(), so , but I was el, reg ') ort ") matrix sion_mod th std cicRegre ecision RandomFonborsCl () and s	ant to ma ression_ del_name of %0.3f ession() TreeClass crestClass assifier td %0.3f	model_: ssifier ssifie ssifier ssifie ssifier ssifie ssifie ssifie	ean, st () r(n_est est_res est_res	d)) ') imators= ult['regult['acc	10) ression_muracy'],	
	# 1. Logistic regres logistic_regres prediction = lo scores = cross_print("%0.2f ac 0.97 accuracy wi Making regression mapproach: best_result = { mean_zer = 0 def generic_reg global mean global best regression_prediction print(regre print(class scores = cr mean = scor std = score if (mean_zer	regression sion = Logi sion.fit (X_ gistic_regression. val_score(1) curacy with the std of Colodels one by ression_models one by ression_model. fit (X_	sticRegres train, y_t ession.pre ogistic_re std of %0 0.02 one like what lel_executi (_train, y_ on_model.pr name,": (y_test, p name,": (y_test, p name,": (y_test) ere(regress leacy'] = me	dict (X_terminate of the strength of the stren	scaled_nescores.mes pove is cute sion_mode est) matrix ' sion_mode est) scion_mode est) scion_mode est) matrix ' sion_mode est) scion_mode es	an(), so , but I was el, reg out ") matrix sion_mod ch std cicRegro candomFonborsCl () and so .026	ant to ma ression_ , y, cv= del_name of %0.3f ession() TreeClas orestCla assifier td %0.3f	model_: ** * (m ** * (b) ** * (c) ** * (d) ** *	ean, st () r(n_est est_res est_res	d)) ') imators= ult['regult['acc	10) ression_muracy'],	
	# 1. Logistic regres logistic_regres prediction = lo scores = cross_print("%0.2f ac 0.97 accuracy wi Making regression mapproach: best_result = { mean_zero = 0 def generic_reg global mean global best regression_prediction print(regre print(confu print(regre print(class scores = cr mean_zero = score if (mean_zer best_re best_re best_re best_re best_re print(regre print() print("= print() list_of_regress list_of_regress list_of_regress list_of_regress list_of_regress list_of_regress list_of_regress list_of_regress for key, value generic_reg print("Best_reg LogisticRegressi [92 0] [1 44]] LogisticRegressi [92 0] [1 44]] LogisticRegressi =	regression sion = Logi sion.fit (X_ gistic_regression.fit (X_ gistic_regression.fit (X_ gistic_regression.fit (X_ curacy with the std of (X_ curacy with the	sticRegres train, y_t ression.pre ogistic_re std of %0 0.02 one like what clear executi clear execu	rain) dict (X_te gression, .2f" % (s tyou see alt on (regres train) edict(X_t confusion classific st, predi ion_model classific st, predi ion_model classific storMachi cmodels.i on(value, ith accur .x report .score 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.9	scaled_nericores.mean cores.mean cove is cute cove is cut	an(), so , but I was el, reg out ") matrix sion_mod ch std cicRegro candomFonborsCl () and so .026	ant to ma ression_ , y, cv= del_name of %0.3f ession() TreeClas orestCla assifier td %0.3f	model_: ** * (m ** * (b) ** * (c) ** * (d) ** *	ean, st () r(n_est est_res est_res	d)) ') imators= ult['regult['acc	10) ression_muracy'],	
	# 1. Logistic r logistic_regres logistic_regres prediction = lo scores = cross_print("%0.2f ac 0.97 accuracy wi Making regression mapproach: best_result = { mean_zero = 0 } def generic_reg global mean global best regression_prediction print(regre print(confu print(regre print(class scores = cr mean = score if (mean_zer best_re best_re best_re print(regre print()) print("=== print()) print("=== print()) print("=== print()) print("=== print()) print("=== print()) print("Best_regress list_of_regress list_of_regress list_of_regress list_of_regress list_of_regress list_of_regress list_of_regress list_of_regress for key, value generic_reg print("Best_reg print("Best_	regression sion = Logi sion.fit (X_ gistic_regr val_score(I curacy with the std of G nodels one by ression_models coresult model.fit (X_ eregression gression_model sion_models fication_ression_model fication_ression_models fication_models fion_models fion_m	sticRegres train, y_t ression.pre ogistic_re std of %0 0.02 one like wha del_executi	rain) dict(X_te gression, .2f" % (s tyou see ab con (regression) .2f" % (s con (regression) .2f" % (scaled_n cores.mea cores.mea cove is cute sion_mode est) matrix ' cation rep ction) cation rep ction rep	an(), so , but I was el, reg out ") matrix sion_mod ch std cicRegro candomFonborsCl () and so .026	ant to ma ression_ , y, cv= del_name of %0.3f ession() TreeClas orestCla assifier td %0.3f	model_: ** * (m ** * (b) ** * (c) ** * (d) ** *	ean, st () r(n_est est_res est_res	d)) ') imators= ult['regult['acc	10) ression_muracy'],	
	# 1. Logistic regres logistic_regres logistic_regres prediction = lo scores = cross print ("%0.2f ac 0.97 accuracy wi Making regression mapproach: best_result = { mean_zero = 0 def generic_reg global mean global best regression_prediction print (regre print (confu print (regre print (confu print (regre print (class score = cr mean_ze best_re print (regre print (regre print (regre print () print ("=== print () print () print ("=== print () pri	regression sion = Logi sion.fit (X_ gistic_regression_fit (X_ grat_score(I) couracy with the std of (I) couracy the std of (I)	sticRegres train, y_t ression.pre ogistic_re std of %0 0.02 one like what cle_executi ctrain, y_ m_model.pr .name,": cy_test, p .name,": confusion classification recall f1 1.00 0.98 0.99 0.99 0.99 0.99 0.99 0.99	rain) dict(X_te gression, 2f" % (s tyou see alt on (regres train) edict(X_t confusion rediction classific st, predi ion_model cl_name'] an %0.3f acc cl_assi restClassi Neighbors st']=Gauss ctorMachi cmodels.i on(value, rith accur at ion report	scaled_n cores.mea cove is cute sion_mode	an(), so , but I was el, reg out ") matrix sion_mo ch std cicRegre cicRegre chorscle and se conditions	ant to ma ression_ , y, cv= del_name of %0.3f reeClas assifier td %0.3f	model_; ssifier ssifie ssifier ssifie ssifier ssifie ssifier ssifie	ean, st () r(n_est est_res est_res	d)) ') imators= ult['regult['acc	10) ression_muracy'],	
	# 1. Logistic regress logistic regres logistic regres prediction = loo scores = cross_print ("%0.2f ac 0.97 accuracy wi Making regression mapproach: best_result = { mean_zero = 0	regression sion = Logi sion.fit (X_ gistic_regression_fit (X_ curacy with the std of (Condels one by ression_models curacy with the std of (Condels one by) ression_models curacy with the std of (Condels one by) ression_model sion_model.fit (X_ curacy with the std of (Condels one by) ression_model sion_model.fit (X_ curacy with the std of (Condels one by) ression_model.fit (X_ curacy with the std of (Condels one by) ression_model.fit (X_ curacy with the std of (Condels one by) ression_model.fit (X_ curacy with the std of (Condels one by) ression_model.fit (X_ curacy with the std of (Condels one by) ression_model.fit (X_ curacy with the std of (Condels one by) ression_model.fit (X_ curacy with the std of (Condels one by) ression_model.fit (X_ curacy with the std of (Condels one by) ression_model.fit (X_ curacy with the std of (Condels one by) ression_model.fit (X_ curacy with the std of (Condels one by) ression_model.fit (X_ curacy with the std of (Condels one by) ression_model.fit (X_ curacy with the std of (Condels one by) ression_model.fit (X_ curacy with the std of (Condels one by) ression_model.fit (X_ curacy with curacy with the std of (Condels one by) ression_model.fit (X_ curacy with curacy with the std of (Condels one by) ression_model.fit (X_ curacy with curacy with the std of (Condels one by) ression_model.fit (X_ curacy with curacy with curacy with the std of (Condels one by) ression_model.fit (X_ curacy with curacy with the std of (Condels one by) ression_model.fit (X_ curacy with curacy with curacy with curacy with the std of (Condels one by) ression_model.fit (X_ curacy with curacy cura	sticRegres train, y_t tession.pre ogistic_re std of %0 0.02 one like wha lel_executi c_train, y_ m_model.pr .name,": cy_test, p .name,": cy_te	rain) dict(X_te gression, 2f" % (s tyou see alt on (regres train) edict(X_t confusion rediction classific st, predi ion_model cl_name'] an %0.3f acc cl_assi restClassi Neighbors st']=Gauss ctorMachi cmodels.i on(value, rith accur at ion report	scaled_n cores.mea cove is cute sion_mode	an(), so , but I was el, reg out ") matrix sion_mo ch std cicRegre cicRegre chorscle and se conditions	ant to ma ression_ , y, cv= del_name of %0.3f reeClas assifier td %0.3f	model_; ssifier ssifie ssifier ssifie ssifier ssifie ssifier ssifie	ean, st () r(n_est est_res est_res	d)) ') imators= ult['regult['acc	10) ression_muracy'],	
[38]:	# 1. Logistic regres logistic_regres prediction = los cores = cross_print("%0.2f ac 0.97 accuracy wi Making regression m approach: best_result = {	regression sion = Logi sion = Logi sion = Sion = Sion = Sion regression = Models one by ression = Models one in the sion	stickegres train, y_t cession.ry tession.re ogistic_re std of %0 0.02 one like what contains y_ memodel.pr name,": cy_test, p name,": confusion classification recall fl non 0.98 0.99 0.99 0.99 0.99 0.99 0.99 0.99	rain) dict(X_tegression, .2f" % (s tyou see alt on (regres train) edict(X_t confusion rediction classific st, predi ion_model cl_name'] an %0.3f acc reeClassi restClass Neighbors st']=Gauss ctomachi models.i on(value, ith accur x reportscore 0.99 0.99 0.99 0.99 0.99 0.99 0.99 0.9	scaled_n cores.mea cove is cute sion_mode	an(), so hot I was hot I was classification matrix sion_mo chandomFi classification candomFi cand	ant to ma ant to ma ression freeClas cores.st ant to ma ression freeClas corestcla assifier td %0.3f	======================================	ean, st () r(n_est est_res est_res	d)) ') imators= ult['reg ult['std	10) ression_m racy'], ']))	odel_name
	# 1. Logistic regres logistic_regres prediction = los cores = cross_print("%0.2f ac 0.97 accuracy wi Making regression m approach: best_result = {	regression sion = Logistic regression = Colored	sticRegres train, y_t ession, y_t ession. y_t ession. re std of %0 0.02 one like wha clel_executi c_train, y_ n_model.pr .name,": c(y_test, p .name,": c(y_test, p .name,": eport(y_te ere(regress cleacy'] = me	rain) dict (X_te gression, .2f" % (s tyou see alt con (regression rediction classific sion_model classific sion_model classific sion report	scaled_n cores.mea cove is cute	an(), so but I was characteristic and sicked and sick	ant to ma ant to ma ant to ma ression ression ression() recclas restclas assifier td %0.3f	incodel_i immodel_i	ean, st ean, st () r(n_est est_res est_res est_res	imators= ult['regult['std	atd. But logingly and the state of the state	istic model neral logistic
	# 1. Logistic r logistic_regres logistic_regres logistic_regres prediction = lo scores = cross_ print("%0.2f ac 0.97 accuracy wi Making regression m approach: best_result = { mean_zero = 0 def generic_reg global mean global best regression_ print(regre print(confu print(regre print(confu print(regre print(confu print(regre print(class scores = cr mean_zer best_re best_re print() print(regre print() print(regre print() print('= print() print('= print() print('= print() print('= print() list_of_regress list_o	regression sion = Logi sion.fit(X gistic_regre val_score(I curacy with the std of (I curacy with	sticRegres train, y_t ression.pre ogistic_re std of %0 0.02 one like wha clear cuti ctrain, y_ m_model.pr name,": cy_test, p name,": confunction confunc	rain) dict (X_te gression, 2f" % (s tyou see ab on (regress train) edict(X_t confusion classific st, predi ion_model classific st, predi ion_model classific st, predi ion_model classific st, predi ion_model classific st, predi ion_model classific st, predi ion_model classific st, predi ion_model classific st, predi ion_model classific st, predi ion_model classific st	scaled_normode set over it is support	an (), see, but I was a contract of the state of the stat	del_name of %0.3f ession() TreeClas orestClas assifier td %0.3f	stethe commodel is sifier assifier assi	ean, st ean, st () r(n_est est_res est_res est_res	d)) ') imators= ult['reg ult['std s rounded ons, I wou	atd. But logingly and the state of the state	istic model heral logistic
	# 1. Logistic r logistic regress logistic regress prediction = lo scores = cross print("%0.2f ac 0.97 accuracy wi Making regression m approach: best_result = { mean_zero = 0 def generic_reg global mean global best regression_ prediction print(regre print(class scores = cr mean = score if (mean_zer print(confu printregre print(class scores = cr mean = score if (mean_zer best_re best_re best_re best_re best_re best_re print(regre print() print('==- print() print('==- print() print('==- print() print('==- print() list_of_regress list_of_regress list_of_regress list_of_regress list_of_regress list_of_regress list_of_regress list_of_regress list_of_regress for key, value generic_reg print("Best_reg LogisticRegressi [92 0] [1 44]] LogisticRegressi [91 1] [6 39]] DecisionTreeClas ===================================	regression sion = Logi sion.fit (X gistic_regression_fit (X gistic_regression_wolds one by ression_model care ression_model sion_models s	sticRegres train, y_t tession.pre ogistic_re std of %0 0.02 one like what clel_executi ctrain, y_ m_model.pr .name,": .name,": .name,": .name,": .report(y_te .refrequess .ssion_mode acy'] = me .ssion_mode acy'] = me .std .name,": .readomFor 'RandomFor 'K-Nearest 'NaiveByte 'Supportor lel_executi lel_executi lel_executi sification recall fl 1.00 0.98 0.99 0.99 0.99 0.99 0.99 0.99	rain) dict(X_te gression, 2f" % (s tyou see als on (regres train) edict(X_te gression) edict(X_te confusion classince strain) edict(X_te confusion classince strain edict(X_te confusion classince strain edict(X_te confusion classince strain edict(X_te confusion classince strain edict(X_te confusion classince classinc	scaled_normal cove is cute co	an(), so an(cores.st ant to ma ression_ ression_ ression() recclas ession() recclas crestClas assifier td %0.3f assifier td %0.3f	act () () () () () () () () () () () () ()	ean, st ean, st () r(n_est est_res est_res cv=10)	d)) ') imators= ult['reg ult['std s rounded ons, I wou	atd. But logingly and the state of the state	istic model neral logistic
	# 1. Logistic regress logistic regress prediction = 10 scores = cross_print("%0.2f ac 0.97 accuracy wi Making regression m approach: best_result = {	regression sion = Logi sion.fit (X gistic_regrevalscore (1 curacy with th std of (0 curacy with	sticRegres train, y_tession.pre consistic_rea std of %0 0.02 one like what consistic_rea std of %0 consistic_rea	rain) dict (X te gression, .2f" % (s tyou see alt on (regress train) edict (X te confusion rediction classific st, prediction rediction classific st, prediction rediction classific st, prediction rediction rediction rediction classific st, prediction rediction red	scaled more scaled	an(), so an(cores.st ant to ma ression re	std 0.0 ice base model_ model_ std 0.0 ice base std 0.0 ice base assifier std 0.0 ice base assifier std 0.0 ice base assifier std 0.0 compared to the compared	aname): ean, st can, st can	alt['regult['accult['std	std. But logingly and state of the state of	istic model neral logistic y pick the